International Journal of Research in Computer Science
eISSN 2249-8265 Volume 2 Issue 2 (2012) pp. 1-5

www.ijorcs.org

ADOPTION OF PARALLEL GENETIC ALGORITHMS FOR THE SOLUTION OF SYSTEM OF EQUATIONS

Shilpa S Babalad[1], Anand M Shivapuji[2]

[1]*Faculty, Dept of Computer Science, CITech, Bangalore, India*
[2]*Research Scholar, CST, Indian Institute of Science, Bangalore, India*

Abstract: - *Analysis and understanding of physical systems require modeling the system as set of simultaneous linear / non-linear equations and generating solutions to satisfy the system of equations. Analytical approach towards solving the system of equations remains practical so long as considerable constraints are imposed on the modeled system to bring in significant simplicity so as to retain the system model within the scope of defined algorithms for solving system of equations. The current work adopts the concept of genetic algorithm towards evolving a solution to a system of equations. The fundamental strength of genetic algorithms lies in the fact the solution generation is practically unconstrained permitting the methodology to become the superset for all possible realizable problems. One of the oft repeated and highlighted / drawbacks of genetic algorithm i.e. requirements of huge initial population and corresponding extended number of computations and hence time is addressed by exploiting multi-processing capabilities of the current generation hardware as well as system software. Adopted strategy for selecting the best population, implementation flow chart along with a case study is presented.*

Keywords: *Evolution, Genetic algorithms, parallel programming, system of equations.*

I. INTRODUCTION

The solution of a system of simultaneous equations is probably one of the most important topics in modern computational engineering [1]. Placing the criticality of simultaneous equations at any level below would be a gross understatement considering the fact that almost all recent ground breaking technological advances are a direct result of increasing ability to solve complex simultaneous equations. Simultaneous equations practically rule the roost in so far as formulating engineering problems in the form of mathematical equations are concerned with the applications spanning such broad areas as computational fluid dynamics [2], structural analysis [3], circuit analysis [4], etc. Solution of a set of simultaneous equations primarily means quantifying the set of variables such that they satisfy all the equations simultaneously [5].

Literature reports several methods towards the solution of simultaneous equations with the simplest being the graphical technique and the Gaussian elimination method [5][6]. As the size of the simultaneous equations increases, recourse is taken to adopting iterative approaches considering that they are more advantageous and less prone to round off errors [7]. Apart from the fact that the iterative approaches are less prone to round off and such errors, a thorough understanding of the physical system helps in making a very good initial guess leading to faster convergence.

The current work primarily focuses on the extension of the generic concept of using “guesses” by adopting Genetic Algorithms (GA) towards solving a set of simultaneous equations. The fundamental idea behind using GA is that a set of equations offering a close to practical representation of any physical system would be difficult to solve using any of the standard established techniques. The primary advantage of GA is that they represent a more scalable choice [8] apart from the fact that they are not under any mathematical constraint or bound by error surface and can solve multi-dimensional, non-differential, non-continuous, and as well as non-parametrical problems. One of the primary drawbacks of GA has been that they are time intensive considering that they spawn a large initial population space (user dependent) and it could be many generations before a solution within the error band actually materializes. Towards overcoming this fundamental drawback, recourse is taken to parallelize segments of the primary code with the focus being mainly on the initial population space and fitness evaluation space. Parallelizing these hot-spots reduces the overall execution time of the code reducing the time footprint of the code.

A sample well defined problems is taken up for investigation and solved using the standard Gauss elimination technique, the Gauss Seidel method [5] and finally using the GA developed for the purpose. The detailed approach and the representative

algorithm towards the methods described are presented in the next section.

II. IMPLEMENTATION APPROACH

The investigation starts off with the formulation of a well defined problem having a definite solution. The case study involved a system of equations with five unknowns and 5 equations. The considered system of equations is of the form as given below in figure 1.

$$\begin{bmatrix} 7 & 3 & 4 & 8 & 8 \\ 6 & 4 & 8 & 7 & 11 \\ 9 & 3 & 6 & 9 & 3 \\ 8 & 12 & 9 & 10 & 5 \\ 7 & 9 & 5 & 2 & 5 \end{bmatrix} \begin{vmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{vmatrix} = \begin{bmatrix} 29 \\ 23 \\ 26 \\ 22 \\ 19 \end{bmatrix}$$

Figure 1. Matrix representation of system of equations

The solution vector for the above defined set of equations is composed of the values X1=2.701, X2=0.009, X3=-1.544, X4=0.806, X5=1.224 with the solutions beings arrived at by adopting the Gaussian elimination method with scaled partial pivoting [9]. The same set of equations is subsequently solved using GA which is formulated as mentioned below;

Genetic algorithm fundamentally relies on the principal of having a randomly generated initial population. The initial population primarily is a column vector representing the potential solution of the system of equations. In GA, many such initial population vectors can be defined and for the present analysis the size of the initial population is taken as 5 column vectors. This is fundamentally a random and arbitrary choice and the 5 columns representing the initial population are as given below.

Table 1. Initial population vectors

V_1	V_2	V_3	V_4	V_5
-0.0935	+1.6509	+1.5275	+1.5275	-0.6326
+1.8911	+0.2291	-0.9091	-0.9091	+1.1320
+0.6578	+0.6302	+0.5089	+0.5089	+0.2794
+1.1521	+1.8153	+1.7267	-0.3792	-0.4224
+2.6221	-0.4632	+0.2722	+0.9258	-0.2686

This initial population is generated using the intrinsic random number generator function with an appropriate seeding function for generating different random numbers each time the code is executed. It is important to note that each time the solution procedure is run, these values could be different and depend on the seed used for generating the random numbers.

The initial population generated is subjected to fitness evaluation as discussed below;

1. Determine a LHS column vector obtained by substituting the first solution vector and multiplying the coefficient matrix with the solution vector.
2. Determine an error column vector by subtracting the LHS column vector with the actual RHS column vector.
3. Determine an error coefficient (EC) which is the sum of square of the individual entries of the error vector.
4. Determine an actual coefficient (AC) which is the sum of square of the individual entities of the RHS vector.
5. The relative accuracy R is defined as

$$R = (AC - EC) / AC$$

The R value generated is compared against the defined accuracy level required. If the accuracy is higher or equal to the required magnitude, then the population vector is directly accepted as the final solution and the program terminates. In the event of the accuracy being lower than the desired accuracy, the next population vector is taken up for evaluation. This follows for the entire population set. In the event of any of the population set satisfying the desired criterion, the same is selected as the final solution and the iteration is terminated. In case none of the 5 initial vectors satisfy the accuracy criterion, the program proceeds to evolve. In this evolution, the solution vector that came closest to the required accuracy in the first generation is carried forward as it is and four more solution vectors are generated randomly. In a random selection, one of the four vectors is selected for cross over and on that vector, by another random selection, a position from one to five is selected and starting from one till the chosen position, the values are replaced using the corresponding values from the second best solution from the previous generation. Thus, a new set of vectors are made available which are assessed for their fitness and in case of no convergence in terms of the satisfaction of the accuracy criterion, a third generation population is generated and so on. This process continues till the system evolves a final solution satisfying the required criterion. The carry forwards of one complete solution from the $(i)^{th}$ iteration to the $(i+1)^{th}$ iteration allows benchmarking of the new set while the cross over permits part carry forward of the goodness of fit from the previous solution.

There are two regions of the program that are independently repetitive i.e. the random number generation and the fitness estimation. The independent repetitiveness permits introduction of parallelization wherein depending on the number of processors on the machine, number of threads are created to handle these two tasks with these two tasks themselves being in series.

Towards implementation of the above described methodology for solution of a system of equations, C programming languages is used on GCC 4.6.1 open source compiler [10] with the parallelization being implemented using OpenMP [11].

III. RESULTS AND DISCUSSIONS

Implementation of the scheme as discussed above allowed solution of the problem presented in a matrix form as in figure 1. Towards assessing the response of the developed algorithm, the same system of equation was solved with different initial population sets one of them indicated in table 1. Table 2 presents the results obtained in five trials for an accuracy requirement of 99% and compares against the actual analytical solution.

Table 2. Evaluated coefficients for the trials

Actual	T-1	T-2	T-3	T-4	T-5
2.701	1.976	2.429	1.813	2.651	2.614
0.009	0.165	0.019	0.104	-0.170	-0.432
-1.544	-0.968	-0.842	-0.651	-0.767	-0.433
0.806	1.355	0.296	0.716	0.589	0.321
1.224	0.723	1.535	1.322	0.885	1.055

Table 3. Comparison of trial solutions

Actual	T-1	T-2	T-3	T-4	T-5
29	27	28	27	27	26
23	22	27	26	23	24
26	27	24	23	27	26
22	26	23	24	23	20
19	17	21	18	19	18

One of the important features that is brought out is the fact that a considerable spread is noticed in the values of the variables but the same does not reflect in terms of any major deterioration in the results. This indicates a rather broad solution space for the system of equations indicating flexibility in the system behavior near the optimal range. The other important feature brought out by the analysis is the differential sensitivity of the coefficients as indicated in figure 2.

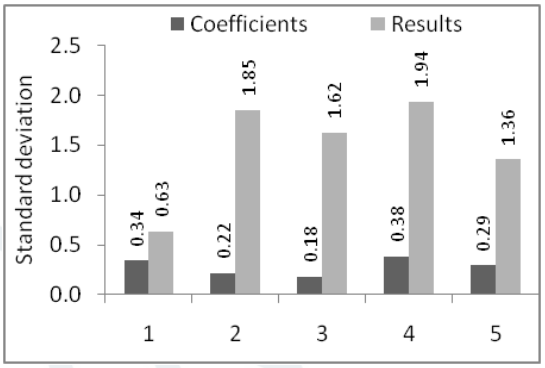

Figure 2. Variation of standard deviation

The system seems to be significantly less sensitive to variable X_1 while being highly sensitive to X_2 and X_3 as evident from the above graph. A multidimensional space with the longest dimension along the X_1 axis would be the solution space for the above system which is one of the most important outcomes of the adoption of GA to the solution of simultaneous equations. In terms of physical systems, this would translate into a significant understanding in terms of the flexibility and restrictions in the considered system since the evaluation has been against a random set of inputs ruling out any pattern and hence bias.

The next investigation pertains to understanding how the choice of initial population affects the number of iterations required for attaining convergence.

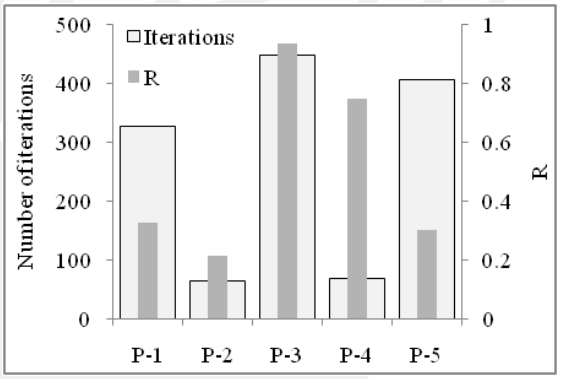

Figure 3. Variation of 'R' and number of iterations

In a very surprising outcome, no real correlation seems to emerge between the quality of the initial population and the number of iterations as evident from figure 3. In case P-3 the R value for initial population is over 0.9 but the number of iteration at about 450 are also the highest among the 5 iterations while P-2 with very low R has the lowest number of iterations. The primary outcome of this analysis is that, the way in which the best chromosomes from the initial population move over to the next generation either full or in part as cross over / mutation needs a thorough analysis and the way this happens could have an impact on the behavior of the GA in general. This aspect of GA as used in the current investigation is being revisited.

Figure 4 deals with the general trend of how the value of R varies with the number of iterations for the best case initial population pertaining to V2 which has only 63 iterations for convergence. As can be seen, through the data seems to be having a general scatter, a clear trend emerges on including a least square fit profile for the data. The value of R tends to increase before finally reaching convergence indicating a gradual improvement with number of iterations. It is the slope of this least square fit that is of importance considering that a higher slope would mean faster rate of improvement in the value of R with increasing number of iterations.

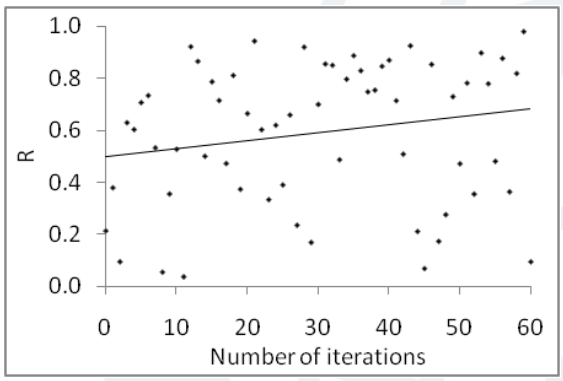

Figure 4. Variation of R with number of iterations

Extending the effort of least square fit to all the five trails conducted, it can be observed that, except for trials T-2 and T-4 all other trials indicated a general degradation of the solution as the number of iterations progressed. The solution obtained is due to pure random behavior and the quality of the previous solution did not have any significant positive impact on the current population. Again, comparing figure 3 and 5, it can be observed that, the number of iterations were within 75 for the case wherein the slope of the curve has been positive i.e. there is a qualitative improvement in the solution over the previous solution. In case where this is not true, it the slope is negative, the number of iterations are significantly high. Trial T-3 seems to have a close to zero slope taking up around 300 iterations while the other two iterations that have greater negative slope require significantly higher number of iterations.

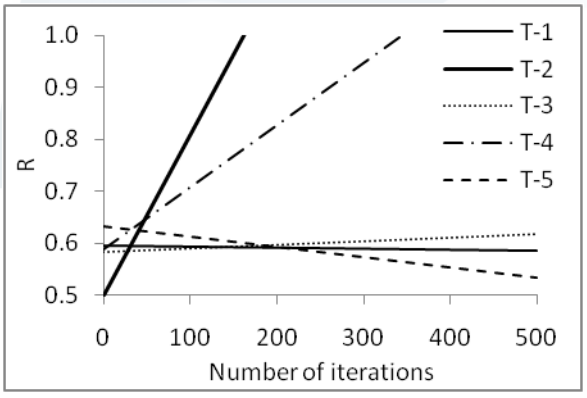

Figure 5. Variation of R with iterations for the five trials

The above results indicate an important trend that though the solution was significantly impacted by the choice of the initial population, it does not depend on the quality of the initial population. This is a rather surprising find and needs more investigation.

Considering the fact that, the number of iterations depends on the choice of the initial population and a particular solution could lead to excessively high number of iterations, parallelization of the hot spots within the code was taken up. The fundamental issue while attempting a comparison between a serial and a parallel code is that since the solution depends on the choice of random numbers, it really is impossible to genuinely compare the two since the random numbers generated in each of the run could be different and could lead to a situation where in the serial code having a better choice of population could outdo the parallel code. The hot spots have however been parallelized using OpenMP and it is believed that the runs are faster as compared to serial code. However no genuine quantification can be provided for the case in question.

IV. CONCLUSIONS

The focus of the current work has been towards the application of parallel genetic algorithms for solving a set of simultaneous equations. A comparison done between the solutions obtained using the above method with the analytical results highlights the ability of GA as a versatile tool capable of handling even non standard system of equations. GA's also provide the solution band / space when a large initial population is considered which can subsequently be curtailed to make the process adaptive in nature. GAs also helps in assessing the sensitivity of the variables. Knowledge of the solution space and the variable sensitivity is a very critical requirement in especially in case of physical systems. GA also indicates the possibility of existence of more than one solution within an error band even when a perfect solution exists. This again is critical in physical systems which operate in a band rather than at a particular point.

On the solution itself, the convergence rate seems to be highly sensitive to the initial population but not in the sense of the accuracy criterion. This is a general anomaly and is being investigated further. This is a very interesting outcome considering the fact that generally if the solution band is known then a system is programmed to be prejudiced to initiate the guess from within the same space. However as the current investigation indicates, this need not be true for all qualification criterion that are designed. Parallelization of hotspots is seen to significantly reduce the time taken for convergence and hence is recommended as a de facto standard to exploit the capabilities of the current generation of hardware and

compilers to reduce the processing times especially while adopting iterative approaches as in the case of genetic algorithm.

V. REFERENCES

[1] Wang D and Zhiming Z, "Differential equations with symbolic computation", Birkhauser Basel, 2005

[2] Versteeg H and Malalasekara W, "An introduction to computational fluid dynamics – The finite volume method", Pearson, 2008

[3] Smith I M and Griffiths D V, "Programming the finite element method", John Wiley & Sons, 2004

[4] Hayt W H, Kemmerly J E and Durbin S M, "Engineering Circuits Analysis", Tata McGraw-Hill Education, 2006

[5] Chapra S and Canale R, "Numerical Methods for Engineers", McGraw-Hill Higher Education, 2009

[6] Hamming R W, "Numerical methods for scientists and engineers", Dover Publications, 1973

[7] Sarkar T, Siarkiewicz K and Stratton R, "Survey of numerical methods for solution of large systems of linear equations for electromagnetic field problems", IEEE Transactions on Antennas and Propagation, Vol, 29, No 6, 1981, pp: 847-856

[8] Gerardo Canfora, Massimiliano Di Penta, Raffaele Esposito, Maria Luisa Villani (2005) "An approach for QoS-aware service composition based on genetic algorithms". Proceedings of the 2005 conference on Genetic and evolutionary computation GECCO 05 (2005).

[9] Markus OLSCHOWKA, "A new pivoting strategy for Gaussian elimination", Linear Algebra and its Applications, Vol 240, 1996, pp:131-151

[10] GNU (2011), "GCC, the GNU Compiler Collection, Version 4.6.1", Available: http://gcc.gnu.org/gcc-4.6/

[11] OpenMP (2011), "The OpenMP API Specification for Parallel Programming, Version 3.1", Available: http://openmp.org/wp/

VI. APPENDIX

The figure below describes the scheme adopted for the solution of system of equations using Genetic Algorithms.